WIKIPEDIA
The Free Encyclopedia

Data conversion

Data conversion is the conversion of computer data from one format to another. Throughout a computer environment, data is encoded in a variety of ways. For example, computer hardware is built on the basis of certain standards, which requires that data contains, for example, parity bit checks. Similarly, the operating system is predicated on certain standards for data and file handling. Furthermore, each computer program handles data in a different manner. Whenever any one of these variables is changed, data must be converted in some way before it can be used by a different computer, operating system or program. Even different versions of these elements usually involve different data structures. For example, the changing of bits from one format to another, usually for the purpose of application interoperability or of the capability of using new features, is merely a data conversion. Data conversions may be as simple as the conversion of a text file from one character encoding system to another; or more complex, such as the conversion of office file formats, or the conversion of image formats and audio file formats.

There are many ways in which data is converted within the computer environment. This may be seamless, as in the case of upgrading to a newer version of a computer program. Alternatively, the conversion may require processing by the use of a special conversion program, or it may involve a complex process of going through intermediary stages, or involving complex "exporting" and "importing" procedures, which may include converting to and from a tab-delimited or comma-separated text file. In some cases, a program may recognize several data file formats at the data input stage and then is also capable of storing the output data in several different formats. Such a program may be used to convert a file format. If the source format or target format is not recognized, then at times a third program may be available which permits the conversion to an intermediate format, which can then be reformatted using the first program. There are many possible scenarios.

Information basics

Before any data conversion is carried out, the user or application programmer should keep a few basics of computing and information theory in mind. These include:

- Information can easily be discarded by the computer, but adding information takes effort.
- The computer can add information only in a rule-based fashion.
- Upsampling the data or converting to a more feature-rich format does not add information; it merely makes room for that addition, which usually a human must do.
- Data stored in an electronic format can be quickly modified and analyzed.

For example, a true color image can easily be converted to grayscale, while the opposite conversion is a painstaking process. Converting a Unix text file to a Microsoft (DOS/Windows) text file involves adding characters, but this does not increase the entropy since it is rule-based; whereas the addition of color information to a grayscale image cannot be reliably done programmatically, as it requires adding new information, so any attempt to add color would require estimation by the computer based on previous knowledge. Converting a 24-bit PNG to a 48-bit one does not add information to it, it only pads existing RGB pixel values with zeroes, so that a pixel with a value of FF C3 56, for example, becomes FF00 C300 5600. The conversion makes it possible to change a pixel to have a value of, for instance,

FF80 C340 56A0, but the conversion itself does not do that, only further manipulation of the image can. Converting an image or audio file in a lossy format (like JPEG or Vorbis) to a lossless (like PNG or FLAC) or uncompressed (like BMP or WAV) format only wastes space, since the same image with its loss of original information (the artifacts of lossy compression) becomes the target. A JPEG image can never be restored to the quality of the original image from which it was made, no matter how much the user tries the "JPEG Artifact Removal" feature of his or her image manipulation program.

Automatic restoration of information that was lost through a lossy compression process would probably require important advances in artificial intelligence.

Because of these realities of computing and information theory, data conversion is often a complex and error-prone process that requires the help of experts.

Pivotal conversion

Data conversion can occur directly from one format to another, but many applications that convert between multiple formats use an intermediate representation by way of which any source format is converted to its target.[1] For example, it is possible to convert Cyrillic text from KOI8-R to Windows-1251 using a lookup table between the two encodings, but the modern approach is to convert the KOI8-R file to Unicode first and from that to Windows-1251. This is a more manageable approach; rather than needing lookup tables for all possible pairs of character encodings, an application needs only one lookup table for each character set, which it uses to convert to and from Unicode, thereby scaling the number of tables down from hundreds to a few tens.

Pivotal conversion is similarly used in other areas. Office applications, when employed to convert between office file formats, use their internal, default file format as a pivot. For example, a word processor may convert an RTF file to a WordPerfect file by converting the RTF to OpenDocument and then that to WordPerfect format. An image conversion program does not convert a PCX image to PNG directly; instead, when loading the PCX image, it decodes it to a simple bitmap format for internal use in memory, and when commanded to convert to PNG, that memory image is converted to the target format. An audio converter that converts from FLAC to AAC decodes the source file to raw PCM data in memory first, and then performs the lossy AAC compression on that memory image to produce the target file.

Lost and inexact data conversion

The objective of data conversion is to maintain all of the data, and as much of the embedded information as possible. This can only be done if the target format supports the same features and data structures present in the source file. Conversion of a word processing document to a plain text file necessarily involves loss of formatting information, because plain text format does not support word processing constructs such as marking a word as boldface. For this reason, conversion from one format to another which does not support a feature that is important to the user is rarely carried out, though it may be necessary for interoperability, e.g. converting a file from one version of Microsoft Word to an earlier version to enable transfer and use by other users who do not have the same later version of Word installed on their computer.

Loss of information can be mitigated by approximation in the target format. There is no way of converting a character like *ä* to ASCII, since the ASCII standard lacks it, but the information may be retained by approximating the character as *ae*. Of course, this is not an optimal solution, and can impact operations like searching and copying; and if a language makes a distinction between *ä* and *ae*, then that approximation does involve loss of information.

Data conversion can also suffer from inexactitude, the result of converting between formats that are conceptually different. The WYSIWYG paradigm, extant in word processors and desktop publishing applications, versus the structural-descriptive paradigm, found in SGML, XML and many applications derived therefrom, like HTML and MathML, is one example. Using a WYSIWYG HTML editor conflates the two paradigms, and the result is HTML files with suboptimal, if not nonstandard, code. In the WYSIWYG paradigm a double linebreak signifies a new paragraph, as that is the visual cue for such a construct, but a WYSIWYG HTML editor will usually convert such a sequence to

, which is structurally no new paragraph at all. As another example, converting from PDF to an editable word processor format is a tough chore, because PDF records the textual information like engraving on stone, with each character given a fixed position and linebreaks hard-coded, whereas word processor formats accommodate text reflow. PDF does not know of a word space character—the space between two letters and the space between two words differ only in quantity. Therefore, a title with ample letter-spacing for effect will usually end up with spaces in the word processor file, for example INTRODUCTION with spacing of 1 em as I N T R O D U C T I O N on the word processor.

Open vs. secret specifications

Successful data conversion requires thorough knowledge of the workings of both source and target formats. In the case where the specification of a format is unknown, reverse engineering will be needed to carry out conversion. Reverse engineering can achieve close approximation of the original specifications, but errors and missing features can still result.

Electronics

Data format conversion can also occur at the physical layer of an electronic communication system. Conversion between line codes such as NRZ and RZ can be accomplished when necessary.

See also

- Character encoding
- Comparison of programming languages (basic instructions)#Data conversions
- Data migration
- Data transformation
- Data wrangling
- Transcoding
- Distributed Data Management Architecture (DDM)
- Code conversion (computing)
- Source-to-source translation

- Presentation layer
- Video Converting (https://v2.youconvert.net/ena/)

References

1. Dragos-Anton Manolescu; Markus Voelter; James Noble (2006). *Pattern Languages of Program Design 5* (https://books.google.com/books?id=9ywUYmxstnAC&q=%22data+conversion%22+%22intermediate+representation%22&pg=PA271). Addison-Wesley Professional. pp. 271–. ISBN 978-0-321-32194-7.

Manolescu, FirstName (2006). *Pattern Languages of Program Design 5*. Upper Saddle River, NJ: Addison-Wesley. ISBN 0321321944.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Data_conversion&oldid=1275770881"